
De latex a EPUB: II-Configuraciones

Víctor Huertas

Licencia CC BY-SA 4.0

(mod: 3 de febrero de 2020) 15 de Diciembre de 2018

Índice

Segunda parte de la serie, donde se aborda la personalización de los documentos HTML obtenidos mediante el comando `htlatex` (o `make4ht` si se prefiere), a través las *configuraciones de usuario* de Tex4ht y los comandos disponibles para este método.

1 Inicio

El *sistema de configuraciones de usuario* para TeX4ht consiste en la utilización de una serie de comandos dentro de un archivo de configuración (con extensión `.cfg`), que es invocado desde el comando `htlatex` del siguiente modo:

```
htlatex midocumento.tex "archivo-cfg,otras-opciones"
```

Se asume que la extensión del archivo es `.cfg`, de modo que sólo se escribe el nombre del archivo en las opciones del comando: esto es, el archivo que utilizaríamos para el comando anterior debe de ser `archivo-cfg.cfg`; además, debe estar localizado en la misma carpeta que el documento `midocumento.tex`. Esta es la forma recomendada de hacerlo, y se desaconseja la utilización de dichos comandos en otros archivos de L^AT_EX, como documentos `.tex` y archivos de estilo `.sty`.

El archivo de configuración debe de seguir el siguiente modelo

```
[Paquetes-requeridos, Definiciones-previas]
\Preamble{opciones}
[Configuraciones-y-definiciones-previas-al-encabezado-HTML]
\begin{document}
[Configuraciones-y-definiciones-para-introducir-dentro-del-encabezado-HTML]
```

```
\EndPreamble
```

En la siguiente sección vamos a ver como funcionan exactamente estas configuraciones dentro del archivo `.cfg`.

2 Funcionamiento de las configuraciones

`TEX4ht` transforma los archivos `LATEX` en HTML mediante la inserción previa de las etiquetas HTML necesarias, en el archivo `.dvi`.

Para realizar dichas inserciones, `TEX4ht` redefine (ya trae prefijadas dichas definiciones, nosotros no tenemos que hacer nada, en principio) los comandos y entornos `LATEX` necesarios y les incorpora una serie de *hooks*. Estos hooks no son más que una suerte de contenedores donde se introducen las etiquetas necesarias para la salida HTML. `TEX4ht` rellena dichos *hooks* con unos valores predeterminados, que pueden ser modificados por el usuario, mediante el comando `\Configure` para el caso de los comandos, y `\ConfigureEnv` para el caso de los entornos. Con detalle, el comando `\Configure` para la configuración `<config-name>` sigue la sintaxis:

```
\Configure{<config-name>}{<hook#1>}{<hook#2>}...{<hook#n>}
```

donde el número de hooks depende del elemento a configurar.

Hay que tener en cuenta que, para la conversión del documento en HTML, por defecto `TEX4ht` sólo maneja adecuadamente los macros `LATEX` cuyo significado ya ha sido declarado mediante las configuraciones que trae predefinidas; y que como ya se ha dicho, podemos modificar los valores por defecto para los hooks mediante el comando `\Configure`.

Para los comandos `LATEX` que `TEX4ht` no reconoce (macros definidos por el usuario o proporcionados por paquetes de terceros), es necesario darles un significado dentro de `TEX4ht` mediante la declaración nuevos hooks; de este modo podrá manejarlos adecuadamente en el proceso de conversión a HTML. Esto se hace mediante el comando `\NewConfigure{configuración}` y se detallará en la parte 3 de la serie.

Para la mayoría de los mortales, la utilización de las configuraciones de `TEX4ht` va a consistir en una tarea artesanal (sí, se trata de un eufemismo para liberar un poco de frustración antes de que el sabio encantador Frestón empiece a facerme encantamientos). `TEX` es un lenguaje de composición tipográfica, bastante distinto a otros lenguajes de programación más familiares e intuitivos, como pueden ser Python, PHP o C, por citar tres ejemplos.

Por la naturaleza tipográfica de `TEX` (o porque Donald Knuth quiso trolearnos), su método de trabajo se basa fundamentalmente en expansiones de macros y en el control de dichas expansiones. Esto lo convierte en un lenguaje cuyo código resulta bastante arduo de leer, y en el que suele ser necesario conocer como funcionan internamente los macros definidos para saber que está haciendo en cada preciso momento; para colmo la documentación de algunos paquetes `LATEX` es bastante precaria. Merece la pena recordar aquí una cita del libro *The TeXBook* de Donald E. Knuth: “[..] es conveniente aprender el concepto pensando en `TEX` como si fuera un organismo vivo”. De aquí que ponerse a inspeccionar el código para ver como trabaja exactamente cierto macro de `TEX4ht` resulte una tarea realmente desesperante y en muchos casos infructuosa (para los mortales).

De este modo, en gran número de casos, la utilización de las configuraciones va a consistir en observar cómo se comporta `TEX4ht` al exportar el documento a HTML, para tomar después nosotros las medidas

oportunas, según nuestras preferencias. También puede ser útil echar un vistazo a las definiciones que `TeX4ht` hace por defecto (en Debian/Ubuntu/Mint, utilizando la distribución TeX Live, se encuentran localizadas en el directorio: `/usr/share/texlive/texmf-dist/tex/generic/tex4ht/`). A todo lo anterior me refería con el término “artesanal”.

Antes de comenzar a detallar las distintas configuraciones, vamos a ver algún pequeño ejemplo para ilustrar brevemente el punto de vista que acabo de exponer.

2.1 Algunas consideraciones y consejos previos

En primer lugar, es conveniente tener en cuenta que a la hora de exportar un documento con `TeX4ht`, es preferible prescindir del mayor número posible de paquetes adicionales que hayamos cargado en el preámbulo. Por ejemplo, debido que yo suelo usar el paquete `titlesec` únicamente a modo decorativo para personalizar los títulos de las secciones, puedo prescindir de dicho paquete cuando voy a exportar a HTML, decorando el documento final mediante estilos CSS. De este modo seremos menos vulnerables a errores fortuitos de compilado (recemos, para que Santa María de Guadalupe, Mística Rosa, nos zarve de la maléfica conjunción `TeX4ht+Babel[spanish]+Hyperref`). También hay que tener presente que `TeX4ht` incorpora configuraciones por defecto específicas para los paquetes `LATEX` más populares, y que en caso de cargar alguno de ellos, dichas configuraciones pueden diferir de las originales, llegando a obtener resultados algo distintos.

Y, por las mismas razones, es preferible usar la menor cantidad posible de macros personalizadas definidos por el usuario (siempre que sea posible).

Además, puesto que puede ser necesario reelaborar todo el documento si se prescinde de ciertos paquetes, sería recomendable planificar condicionalmente los distintos elementos que lo forman, teniendo en cuenta que paquetes, configuraciones y comandos van a ser utilizados, según estemos compilando a PDF o transformando en HTML. Dejo este apartado para otro momento, al que intentaré dedicar un artículo más conciso.

Otro ejemplo de este tipo de situaciones: supongamos que hemos usado en repetidas ocasiones el comando `\rule` en el documento, y que al exportarlo a HTML mediante `TeX4ht`, no queda estéticamente decente (ya te puedo asegurar yo que va a ser así); en dicho caso podemos redefinir (¿indefinir?) en el archivo de configuración el macro `\rule` del siguiente modo:

```
\renewcommand\rule[2]{}{}
```

Y si no ocurre ningún conflicto interno con algún otro comando de `TeX` que utilice el macro `\rule`, habremos hecho desaparecer las líneas del documento HTML obtenido mediante `TeX4ht`.

2.2 Ejemplos Iniciales

Vamos a romper el hielo con una configuración para el texto en negrita; para ello se usa una instrucción que seguirá el esquema:

```
\Configure{textbf}{<hook#1>}{<hook#2>}
```

donde `<hook#1>` es el texto en HTML que se insertará precediendo el texto que en `LATEX` habíamos especificado en negrita y `<hook#2>` es el texto que se insertará después del mismo. Por tanto podemos

utilizar la configuración

```
\Configure{textbf}{\NoFonts\HCode{<strong>}}{\HCode{</strong>\EndNoFonts}}
```

De este modo el fragmento de texto \LaTeX "... all `\textbf{work}` and no play ...", quedará convertido en HTML como

```
... all <strong>work</strong> and no play ...
```

Para dar una idea, en esta configuración (recuerdo que los distintos comandos y configuraciones utilizados serán debidamente relatados en la siguiente sección), `\NoFonts` evita que $\TeX4ht$ decore de manera automática el formato de las fuentes, con los estilos que utiliza por defecto (lo que hace mediante molestas etiquetas ``, que yo prefiero evitar, a ser posible). Y `\HCode{}` se emplea para escapar distintos elementos (conflictivos dentro del código \LaTeX) del estándar HTML.

Es buen momento para hablar del uso de ciertas expresiones y construcciones que se utilizan en las configuraciones $\TeX4ht$, que pueden hacer más confuso su funcionamiento exacto, pero que considero preferible introducirlas desde el comienzo por ese mismo motivo.

En el caso de la configuración de tipo de fuente, es posible encontrar en artículos y páginas web acerca de $\TeX4ht$, alguna configuración similar a la siguiente:

```
\Configure{textbf}{%
  \ifvmode\ShowPar\fi\NoFonts\HCode{<strong>}%
}{%
  \HCode{</strong>\EndNoFonts}%
}
```

"`\ifvmode`", es un comando \TeX que verifica el modo de trabajo en el que se encuentra en dicho momento (*vertical mode*, *horizontal mode*, *math mode* y otros cuantos más). Y el comando `\ShowPar` se utiliza para indicar a $\TeX4ht$ que muestre el código HTML del siguiente párrafo (que a su vez se configuran con `HtmlPar`, que se verá más adelante).

Cuando introduce el primer token de un párrafo, \LaTeX pasa de inmediato a modo horizontal. Por tanto, si el texto en negrita está en mitad del párrafo, no se estará en modo vertical en dicho momento, y no llegará a actuar el comando `ShowPar`. Por tanto, por un lado, el comando `ShowPar` sólo actuará cuando el texto en negrita está justo al principio del párrafo; pero por otro lado, el funcionamiento por defecto de $\TeX4ht$ es mostrar el código de párrafo HTML para un párrafo \LaTeX .

Así que, en condiciones normales, el comando `\ShowPar` no actuará cuando el texto en negrita está al comienzo del párrafo. De hecho, observemos que la configuración por defecto (en el archivo `html4.4ht`) no utiliza `\ShowPar`:

```
\Configure{textbf}{%
  \Protect\HCode{<span class="textbf">}\NoFonts%
}{%
  \EndNoFonts\Protect\HCode{</span>}%
}
```

Sin embargo, a menudo suele emplearse por omisión, construcciones del tipo `\par\ShowPar`, `\ifvmode\IgnorePar\fi\EndP`, para asegurarse que el párrafo será mostrado, por si existe alguna otra

instrucción previa donde el comportamiento por defecto solicite ignorar el párrafo siguiente (por ejemplo, si se incluyó, por precaución, la sentencia `Ignorepar` en algún punto previo, para evitar la posible inserción de un párrafo no deseado, y ésta sigue activa porque finalmente \LaTeX no introdujo ningún párrafo).

Otra expresión muy utilizada al inicio de la inserción de las etiquetas HTML en las configuraciones es `“\ifvmode\IgnorePar\fi\EndP”`. Mediante el comando de `“\EndP”`, se le indica a \TeX4ht que cierre el párrafo actual (si es que está todavía abierto) sobre el que se está trabajando. Por otro lado `“\IgnorePar”` solicita que se omita la inserción del siguiente párrafo (como ya he avanzado).

Si queremos que el código HTML, para el macro que estamos configurando, no se encuentre encerrada en un párrafo HTML, tiene sentido utilizar la expresión anterior, aunque sólo sea por precaución: si \LaTeX se encuentra en modo vertical (apilando párrafos) es que ya ha terminado el párrafo anterior, y por tanto nos interesa que \TeX4ht ignore el párrafo siguiente que \LaTeX abrirá; si por el contrario, este último se encuentra en medio de un párrafo, conviene indicar a \TeX4ht que lo cierre, antes de añadir el código HTML en cuestión (y \TeX4ht no abrirá nuevo párrafo hasta que \LaTeX vuelva a hacer lo propio, o lo hagamos nosotros manualmente).

Vamos a aplicar las expresiones anteriores a otro ejemplo: supongamos que queremos configurar la instrucción de seccionado `\paragraph` para las secciones-párrafo de \LaTeX (voy a utilizar este término compuesto para no confundirlo con cada uno los párrafos de texto).

```
\Configure{paragraph}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}
```

En primer lugar tenemos una dificultad intrínseca del seccionado en \LaTeX : no está muy claro donde termina un párrafo (el `paragraph` del seccionado no se corresponde con un párrafo físico de texto); probablemente lo haga antes del inicio de la siguiente instrucción de seccionado; por este motivo no voy describir los dos primeros *hooks*. Respecto a los otros, `<hook#3>` contiene el código que se insertará previo al título de párrafo, y `<hook#4>` contendrá el código que se insertará después del título del mismo.

También conviene aclarar que no es sencillo, sin un conocimiento exhaustivo del código de \TeX4ht , establecer la relación exacta entre los párrafos introducidos por éste último a partir de los párrafos de \LaTeX . De hecho, puede depender de la Configuración utilizada, o de la misma definición de la Configuración. Como ejemplo, la configuración del entorno `verbatim` introduce manualmente un párrafo al final, como se puede observar en el código siguiente (extraído también de `html4.4ht`):

```
\ConfigureEnv{verbatim}{\env:verb{verbatim}}{\endenv:verb}{}{}
\ConfigureEnv{verbatim*}{\env:verb{verbatim}}{\endenv:verb}{}{}
\HAssign\verbatimN 0
\def\env:verb#1{\ifinner\else \vfill\break\fi
\IgnorePar\EndP \HAssign\Verb:N=0
\gHAdvance\verbatimN by 1
\HCode{<div class="#1" id="verbatim-verbatimN"\a:LRdir>}\EndNoFonts}
\def\endenv:verb{\NoFonts
\EndP \ht:special{t4ht=</div>}\ShowPar \par{\HCondtrue\noindent}%
}
```

Por defecto, \TeX4ht transforma una sección-párrafo

```
[...]
\paragraph{Mi título del párrafo}
All work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy,
all work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy.

All work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy,
all work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy.

\section{Siguiente sección}
[...]
```

del siguiente modo (que escribo de forma genérica):

```
<p class="indent" > <span class="paragraphHead"><a
  id="xxx"></a><span
class="xxx">Titulo de Parrafo.</span></span>
All work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy,
all work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy.
</p>

<p>All work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy,
all work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy.
</p>

<h3 class="sectionHead"><span class="titlemark">X. </span> <a
  id="xxx"></a>Siguiente sección</h3>
```

Observemos que el título del `\paragraph`, junto con su primer fragmento de texto que viene a continuación, forman un sólo párrafo HTML (como también ocurre en \LaTeX).

Supongamos que, como objetivo, queremos que la salida en HTML utilice etiquetas de encabezado, siguiendo la forma siguiente:

```
<h4>Titulo del Párrafo</h4>
<p class="indent" >
All work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy,
all work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy.
</p>
```

Si decidimos probar inicialmente la configuración (que voy a llamar **config-1**):

```
\Configure{paragraph}{\HCode{<h4>}}{\HCode{</h4>}}
```

veremos que no es adecuada, pues obtenemos la siguiente incorrecta estructura HTML (en realidad, por el complejo funcionamiento de $\text{T}_{\text{E}}\text{X}4\text{ht}$, el resultado real suele ser algo más desaliñado, con espacios, saltos de línea y líneas en blanco por aquí y por allá)

```
<p class="indent" > <h4><a id="xxx"></a>Titulo de Parrafo.</h4>
All work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy,
all work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy.
</p>
```

Parece que va a ser necesario cerrar el párrafo antes de abrir la etiqueta para el encabezado. Veamos que sucede (voy a actuar como alguien que no supiera lo que va a pasar) si añadimos el comando $\backslash\text{EndP}$ (para cerrar el párrafo) previamente; algo como la siguiente configuración (que voy a llamar **config-2**):

```
\Configure{paragraph}{\EndP\HCode{<h4>}}{\HCode{</h4>}}
```

Para nuestra sorpresa, el resultado obtenido sigue siendo el mismo.

Veamos como, en términos aproximados (yo no puedo hacer más), estaría funcionando $\text{T}_{\text{E}}\text{X}4\text{ht}$ en la configuración **config-1**: por limitaciones intrínsecas del diseño del mecanismo de trabajo de $\text{T}_{\text{E}}\text{X}$, parece que $\text{T}_{\text{E}}\text{X}4\text{ht}$ cierra un párrafo HTML justo en el momento de abrir el siguiente (esto es algo que se intuye de la configuración HtmlPar , que se verá en la siguiente sección).

Normalmente, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ empieza nuevo párrafo con cada instrucción de seccionado, y hará lo propio con $\backslash\text{paragraph}$. De este modo, antes de comenzar con la inserción del título del comando $\backslash\text{paragraph}$, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ se encuentra en modo vertical. Cuando se llega a la primera inserción de texto en la configuración para $\backslash\text{paragraph}$ (lo que viene antes del propio título, y en este caso es $\backslash\text{HCode}\{<h4>\}$), se abre el nuevo párrafo y $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ pasa de inmediato a modo horizontal; entonces $\text{T}_{\text{E}}\text{X}4\text{ht}$, cierra el párrafo HTML anterior y abre uno nuevo, el que contendrá el citado texto.

Es así como se aprecia que, efectivamente, el comando $\backslash\text{EndP}$ añadido en la configuración **config-2**, no va a ser de utilidad: se fuerza a terminar el párrafo HTML, correspondiente al párrafo $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ previo a la inserción del comando $\backslash\text{paragraph}$, y se abrirá uno nuevo después. Pero observemos que esto es justo lo que ya sucedía de forma automática con la configuración **config-1**.

Lo que en realidad necesitamos en dicho punto es que $\text{T}_{\text{E}}\text{X}4\text{ht}$ ignore el **siguiente** párrafo, el que abriría con $\backslash\text{HCode}\{<h4>\}$, lo que haríamos con el comando $\backslash\text{IgnorePar}$. Además no debería de ser necesario utilizarlo junto con $\backslash\text{ifvmode}$, puesto que $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ empieza nuevo párrafo con las instrucciones de seccionado (como ya se ha comentado), pero por comodidad insertaremos esa construcción estándar". La configuración quedaría

```
\Configure{paragraph}{\ifvmode\IgnorePar\fi\EndP\HCode{<h4>}}{\HCode{</h4>}}
```

Pero vemos que no es suficiente cuando obtenemos un resultado como el siguiente (que adecento un poco para que se vea mejor su estructura)

```
<h4><a id="xxx"></a>Titulo de Parrafo.</h4>
All work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy,
all work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy.
<p class="indent">
All work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy,
all work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy.
</p>
```

donde, como podemos ver, \TeX4ht deja de insertar el párrafo HTML correspondiente a lo que debería de ser el párrafo posterior al título de la sección-párrafo.

Hay que tener en cuenta que, \TeX4ht no abrirá de forma automática nuevo párrafo HTML, hasta que \LaTeX cierre el actual y abra otro; y como ya se ha explicado previamente, las secciones-párrafo de \LaTeX engloban el propio título, de modo que no comenzará nuevo párrafo hasta después de haber concluido la sección-párrafo. Si se quiere volver a empezar párrafo HTML después del título de la sección-párrafo (tras haber empleado una instrucción \IgnorePar en \TeX4ht), sólo hay que insertar un párrafo \LaTeX , \par en la configuración, del siguiente modo:

```
\Configure{paragraph}{}{%{ %
  \IgnorePar\HCode{<h4>} %
}{ %
  \HCode{</h4>}\ShowPar\par %
}
```

obteniendo el resultado deseado

```
<h4><a id="xxxx"></a>Titulo de Parrafo.</h4>
<!--l.III --><p class="indent" >
All work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy,
all work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy.
</p>
<p class="indent">
All work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy,
all work and no play makes Jack a dull boy, all work and no play
makes Jack a dull boy, all work and no play makes Jack a dull boy.
</p>
```

Por precaución, he usado la construcción $\text{\ShowPar}\text{\par}$, aunque en esta sencilla construcción parece que \IgnorePar siempre afectará al párrafo que \LaTeX comienza al introducir \h4 , no pudiendo tener efecto para párrafos posteriores.

Conviene aclarar, también, que la forma invertida `\par\ShowPar` (que he visto utilizar en no pocas ocasiones) parece funcionar exactamente igual, es decir, siempre que comando `ShowPar` se introduzca inmediatamente después de la instrucción de párrafo.

En las siguientes secciones vamos a presentar las principales configuraciones y comandos que se pueden utilizar en los archivos de configuración, agrupados por categorías, con algunos ejemplos ilustrativos. En la parte 3 puede que describa otras configuraciones más específicas, como las relacionadas con las tablas de contenidos y las referencias cruzadas (que yo apenas he utilizado).

3 Configuraciones más destacadas

3.1 Envoltura/Contenedor del documento

Configuraciones

- `\Configure{PROLOG}{<hook#1>}`:

Se utiliza para especificar los elementos de la cabecera, previos a la etiqueta HTML, que aparecerán en el documento creado, como el tipo de documento (DOCTYPE) o la versión XML. Resulta útil si se desea eliminar rápidamente algunos de dichos elementos que vienen por defecto, según las opciones de conversión con que se ha utilizado el comando `htlatex`.

Por ejemplo, con las opciones “`xhtml,mathml`” (utilizando la codificación utf-8), el encabezado por defecto que crea `htlatex` es:

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0//EN"
"http://www.w3.org/Math/DTD/mathml2/xhtml-math11-f.dtd" >
<html xml:lang="es" xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title>De latex a EPUB: II-Configuraciones</title>

  [...]
```

Si queremos que no se creen las líneas para la versión XML y el DOCTYPE, sólo hay que introducir en el archivo de configuración:

```
\Configure{PROLOG}{}
```

Y el inicio del archivo HTML quedaría (de forma no válida, claro está):

```
<html xml:lang="es" xmlns="http://www.w3.org/1999/xhtml"
>
<head>
  <title>De latex a EPUB: II-Configuraciones</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="generator" content="TeX4ht (http://www.tug.org/tex4ht)" />
<meta name="originator" content="TeX4ht (http://www.tug.org/tex4ht)" />
```

- `\Configure{DOCTYPE}{<hook#1>}`: Se utiliza para introducir el código que conformará la declaración de tipo de documento.

Cada una de las cuatro siguientes configuraciones marcan la ubicación natural para las correspondientes etiquetas de apertura (primer hook) y cierre (segundo hook), según la estructura de una página HTML.

- `\Configure{HTML}{<hook#1>}{<hook#2>}`.
- `\Configure{BODY}{<hook#1>}{<hook#2>}`.
- `\Configure{HEAD}{<hook#1>}{<hook#2>}`.
- `\Configure{TITLE}{<hook#1>}{<hook#2>}`.
- `\Configure{TITLE+}{<hook#1>}`: Se utiliza para introducir el texto del título de la página HTML. No funciona (al menos en las versiones de \TeX 4ht por defecto en las últimas distribuciones estables de Debian/Ubuntu/LinuxMint)
- `\Configure{@HEAD}{<Hook#1>}`: Añade el contenido de `<Hook#1>` al contenido de la cabecera del documento HTML, justo después de insertar el contenido `<hook#1>` de la configuración para HEAD. Además el comando es *aditivo*: se puede usar repetidas veces, de modo que los respectivos contenidos de cada comando se irán añadiendo consecutivamente al encabezado.
- `\Configure{@BODY}{<Hook#1>}`: Análogo a `@HEAD` pero para la etiqueta `<body>` y la configuración BODY.

Comandos y entornos

`\Preamble{}` ... `\EndPreamble`: Delimita el contenido cuya configuración se efectuará antes de comenzar a introducir el encabezado HTML.

`\ifOption`: Sirve para verificar que opciones han sido introducidas durante la ejecución del comando `htlatex`. Tiene la forma general

```
\ifOption{<opción-a-comprobar>}{<hacer-si-usada>}{<hacer-si-no-usada>}
```

3.2 Hojas de estilo CSS

3.2.1. Comandos

- `\Css{<instrucciones-css>}`: Añade el código CSS insertado mediante dicho comando al contenido del archivo CSS generado automáticamente (de nombre `\jobname.css`) por \TeX 4ht.

3.3 Título, seccionamiento, links para secciones y tabla de contenidos

Configuraciones

- `\Configure{maketitle}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}`: Configura la *página de título* (título, autor, fecha,...); el contenido de `<hook#1>` y `<hook#2>` son insertados antes y después de la

página de título respectivamente. `<hook#3>` y `<hook#4>` son insertados antes y después, respectivamente, del título del documento.

- `\Configure{thanks author date and}{<h#1>}{<h#2>}{<h#3>}{<h#4>}{<h#5>}{<h#6>}{<h#7>}{<h#8>}`: Se utiliza para configurar los agradecimientos, el autor, la fecha y algunos otros propósitos. Los hooks son introducidos de la siguiente manera:

- (i) `<h#1>`: antes de los agradecimientos;
- (ii) `<h#2>`: después de los agradecimientos;
- (iii) `<h#3>`: antes del autor;
- (iv) `<h#4>`: después del autor;
- (v) `<h#5>`: antes de la fecha;
- (vi) `<h#6>`: después de la fecha.

Las siguientes cinco configuraciones siguen un mismo patrón:

- `\Configure{section}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}`.
- `\Configure{subsection}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}`.
- `\Configure{subsubsection}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}`.
- `\Configure{paragraph}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}`.
- `\Configure{subparagraph}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}`.

Donde `<hook#1>` es insertado antes de comenzar la sección, y `<hook#2>` es insertado al finalizar la misma; `<hook#3>` es insertado previo al título de la sección, y `<hook#4>` tras concluir el título.

Las configuraciones correspondientes a las secciones estrelladas siguen el mismo esquema, designadas con el mismo nombre precedido de `like`: `likesection`, `likesubsection`, `likesubsubsection`, `likeparagraph`, `likesubparagraph`.

3.4 Tabla de Contenidos

Configuraciones

Para modelar la apariencia de la tabla de contenidos se utiliza la siguiente configuración

- `\Configure{tableofcontents}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}{<hook#5>}`:
 - (i) `<hook#1>` es introducido antes de la tabla de contenidos;
 - (ii) `<hook#2>` es insertado justo antes de finalizar la misma;
 - (iii) `<hook#3>` es introducido después de finalizar la tabla de contenidos;
 - (iv) `<hook#4>` es insertado antes de un párrafo *no-indentado*;
 - (v) `<hook#5>` es colocado antes de un párrafo *indentado*.

Como se puede apreciar, la configuración para `tableofcontents` no permite personalizar el título de la tabla de contenidos (desconozco ahora mismo si existe una configuración específica para ello). En el siguiente ejemplo empleamos un truco para modificar el estilo y el contenido del título de la tabla de contenidos (por defecto, en español el título de la tabla de contenidos es “Índice”):

Ejemplo:

```
\let\contentsname\empty
\Configure{tableofcontents}
{ %
  \ifvmode\IgnorePar\fi\EndP %
  \HCode{ %
    </div>\Hnewline<nav id="nav-toc" class="column">\Hnewline<h2>Tabla de %
    Contenidos</h2>\Hnewline<ul>%
  } %
}
{}
{ %
  \ifvmode\IgnorePar\fi\EndP %
  \HCode{ %
    </ul>\Hnewline</nav>\Hnewline<div id="aux-content-3" class="column">%
  } %
}
{}{}
```

Comandos y entornos

El siguiente comando se utiliza para configurar cada una de las entradas de la tabla de contenidos, por secciones, subsecciones, etc.

- `\ConfigureToc{<unidad>}{<delante-marca>}{<delante-título>}{<delante-num-pág>}{<final>}`

Ejemplo: Para poner cada entrada para un sección en un párrafo se puede utilizar la siguiente configuración

```
\ConfigureToc{section}{ }{\NoFonts\HCode{<p>}}{\HCode{</p>}\EndNoFonts}{ }
```

Además, si para cierta unidad se dejan vacíos todos los elementos, dicha unidad no aparecerá en la tabla de contenidos.

3.5 Párrafos

Configuraciones

`\Configure{HtmlPar}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}`: Configura el formato que `TEX4ht` utilizará para abrir y cerrar los párrafos.

El primer hook contiene el código (generalmente HTML, recordemos que `TEX4ht` también puede exportar a otros formatos, como ODF) que será insertado por `TEX4ht` para el comienzo de un párrafo `LATEX` no indentado. `<hook#2>` es análogo al primero, pero para párrafos `LATEX` indentados.

`<hook#3>` contiene el código que será introducido al insertar el comando de cierre `\EndP`, para párrafos \LaTeX no indentados. El último hook es análogo al tercero pero para párrafos indentados.

La configuración por defecto (para un artículo \LaTeX genérico) sigue el esquema (en realidad lo he simplificado, pues la versión original registra los números de líneas y atiende a otras cuestiones relevantes, resultando más compleja de lo que se muestra a continuación):

```
\Configure{HtmlPar}{%
  \EndP \HCode{<p class="noindent">}%
}{%
  \EndP \HCode{<p class="indent">}%
}{%
  \HCode{</p>}%
}{%
  \HCode{</p>}%
}
```

En el ejemplo se aprecia la peculiar circunstancia, antes mencionada: \TeX4ht cierra el párrafo sólo en el momento en que se va a abrir uno nuevo.

Hay que tener en cuenta que se trata de una configuración básica, que es aconsejable no modificar en caso de duda.

Comandos y entornos

- `\IgnorePar`: Solicita a \TeX4ht que ignore el siguiente párrafo de \LaTeX .
- `\ShowPar`: Pide a \TeX4ht que muestre (no ignore) el siguiente párrafo \LaTeX .
- `\HtmlParOff [...]\HtmlParOn`: Deshabilita la inserción de párrafos para el código introducido entre ellos.
- `\EndP`: Solicita a \TeX4ht que cierre el párrafo (HTML) actual (lo que hará de acuerdo a la configuración introducida para `HtmlPar`).
- `\ShowIndent`: Requiere a \TeX4ht que indente el siguiente párrafo.
- `\IgnoreIndent`: Solicita a \TeX4ht que no indente el siguiente párrafo.

3.6 Comandos especiales

Los siguientes comandos son necesarios para formatear el texto HTML introducido en las configuraciones:

- `\HCode`: Permite introducir las etiquetas HTML sin necesidad de escapar (mediante ningún otro método) los signos “<”, “>”.
- `\Tg`: Tiene la misma funcionalidad `\HCode`, . Nunca lo he usado, porque daba problemas hace algún tiempo (no se si lo habrán corregido) y se puede usar `\HCode`.
- `\Hnewline`: Produce un salto de línea dentro del código fuente HTML (no afectará al visualizado, en principio). Se utiliza dentro del comando `\HCode`.

- `\HChar`: Se utiliza para mostrar el carácter ASCII correspondiente al número entero introducido. Por ejemplo, teniendo en cuenta que los códigos ASCII para los caracteres “<”, “>” son 60 y 62 respectivamente, se puede configurar el texto en negrita de la siguiente forma:

```
\Configure{textbf}{\NoFonts\HChar{60}strong\HChar{62}}{\HChar{60}/strong%
\HChar{62}\EndNoFonts}
```

3.7 Configuración de entornos: `\ConfigureEnv`

3.8 Imágenes

Configuraciones

- `\Configure{Picture}{<hook#1>}`: Configura la extensión (atención: sólo la extensión, no el formato de archivo de imagen) de los archivos de imagen producidos por los comandos `\Picture+` y `\Picture*`. En `<hook#1>` se especifica la extensión de imagen.
- `\Configure{Picture+}{<hook#1>}{<hook#2>}`: Se utiliza para configurar el entorno `Picture+ ... \EndPicture` (ver subsección posterior), donde el código de `<hook#1>` será insertado antecediendo al contenido del entorno y `<hook#2>` después del mismo (al parecer solo configura el entorno `\Picture+{}` y no `\Picture+[]{}`).
- `\Configure{Picture*}{<hook#1>}{<hook#2>}`: Exactamente igual que el anterior pero para el entorno `Picture*` (al parecer solo configura `\Picture*{}` y no `\Picture*[]{}`).
- `\Configure{IMG}{<h#1>}{<h#2>}{<h#2>}{<h#4>}{<h#5>}`: Permite configurar al etiqueta HTML (en general, `img`, para mostrar la imagen) con la que se incluirán las imágenes en el documento, donde
 - (i) `h#1` es insertado antes del nombre de archivo de la imagen;
 - (ii) `h#2` es insertado entre el nombre de archivo de la imagen y el valor del atributo `alt`;
 - (iii) `h#3` es insertado como cierre para el valor del atributo `alt`, cuando se trata de una imagen generada utilizando `\Picture+` o `\Picture*`;
 - (iv) `h#4` es insertado como cierre para el valor del atributo `alt`, cuando se trata de una imagen generada mediante `\Picture`;
 - (v) `h#5` es insertado al finalizar.

Comandos y entornos

- `Picture`: Comando que sigue la forma

```
\Picture [<valor_atributo_alt>]{<ruta_imagen>}\EndPicture
```

y se utiliza para insertar una imagen en el documento.

- `Picture+`: Entorno con la forma

```

\Picture+ [<valor_atributo_alt>]{<nombre_imagen><espacio><otros_atributos>}
<texto_a_convertir_a_imagen>
\EndPicture

```

que genera una una imagen del texto encerrado, incluyéndola en el documento con un visualizado “inline”.

<nombre_imagen> es utilizado para guardar la imagen creada con dicho nombre. Si no se quiere especificar ningún nombre de imagen, es necesario dejar el espacio que separa el nombre de imagen de los otros atributos; de ese modo, T_EX4ht nombrará la imagen automáticamente. Es resto de atributos deben ser especificados separados por espacios de la forma

```
atributo#1=valor#1 ... atributo#n=valor#n
```

no encerrando entre comillas los valores de los atributos (lo que conllevaría errores de compilado); T_EX4ht ya se encargará de colocarlas automáticamente.

- **Picture***: Variante del entorno **Picture+**, que permite generar una imagen del texto encerrado, incluyéndola en el documento con un visualizado “en bloque” (esto último según la teoría, pues en la práctica yo no he apreciado ninguna diferencia; introduce la imagen dentro de un párrafo HTML, del mismo modo que hace **Picture+**. Todavía no tengo muy claro si son diferentes por sí mismos, o si por convención se aconseja configurarlos, uno para visualizado “en bloque” y el otro “inline”).

Ejemplo:

Si aplicamos las configuraciones:

```

\Configure{Picture*}{%
  \ifvmode\IgnorePar\fi\EndP\HCode{<div class="text-picture ">}%
}{%
\ifvmode\IgnorePar\fi\EndP\HCode{</div>}%
}%
%
\Configure{IMG}{%
  \HCode{<img src="" %
}{%
  \HCode{" alt="" %
}{%
  "%
}{%
  \HCode{"} %
}{%
  \HCode{>} %
}

```

el ejemplo `\Picture*{text-pic}All work and no play\EndPicture` dará como resultado el código

```

<div class="text-picture ">
</div>

```

3.9 Fuentes

Las siguientes configuraciones se utilizan para especificar el estilo de las fuentes en el documento HTML:

- `\Configure{texttt}{<hook#1>}{<hook#2>}`: Configuración para el tipo de fuente *máquina de escribir*.
- `\Configure{textit}{<hook#1>}{<hook#2>}`: Configuración para el tipo de fuente *itálica*.
- `\Configure{textrm}{<hook#1>}{<hook#2>}`: Configuración para el tipo de fuente redondilla.
- `\Configure{textsl}{<hook#1>}{<hook#2>}`: Configuración para el tipo de fuente *inclinada*.
- `\Configure{textsf}{<hook#1>}{<hook#2>}`: Configuración para la familia de fuentes sans serif (sin línea de pie).
- `\Configure{textup}{<hook#1>}{<hook#2>}`: Configuración para el tipo de fuente vertical.
- `\Configure{textbf}{<hook#1>}{<hook#2>}`: Configuración para el tipo de fuente **negrita**.
- `\Configure{textsc}{<hook#1>}{<hook#2>}`: Configuración para el estilo de letra VERSALITA.
- `\Configure{emph}{<hook#1>}{<hook#2>}`: Configuración para el estilo de letra *resaltado*.

Todos las configuraciones adminten dos *hooks*: el primero será introducido precediendo al texto al que se le aplica el estilo de fuente, y el segundo es introducido a continuación del mismo.

Veamos un ejemplo particular, para configurar el texto en negrita; la configuración

```
\Configure{textbf}{\NoFonts\HCode{<strong>}}{\HCode{</strong>}\EndNoFonts}
```

generará una salida HTML para el código L^AT_EX, `\textbf{All work and no play}`, similar a la siguiente

```
<strong>All work and no play</strong>
```

Como curiosidad, se puede destacar que T_EX4ht, aparéntemente, carece de configuraciones para manejar el tamaño de las fuentes (`tiny`, `scriptsize`, `big`, `large`, ...); desconozco el motivo.

3.10 Teoremas

Para modificar la configuración de los teoremas se puede utilizar la siguiente configuración:

```
\Configure{newtheorem}{<hook#1>}{<hook#2>}{<hook#3>}
```

Si añadimos un nuevo teorema a nuestro texto L^AT_EX mediante las instrucciones

```
% En el preámbulo
\newtheorem{\miteorema}[<contador>]{Mi-Título-del-Teorema}
[...]
% En el cuerpo del documento
\begin{\miteorema}
```



```
<Cuerpo del teorema>
\end{miteorema}
```

entonces, `<hook#1>` es insertado antes de `\begin{miteorema}`, `<hook#2>` es intercalado entre el título del teorema y el cuerpo del mismo, y `<hook#3>` es incluido después de `\end{miteorema}`.

Ejemplo: mediante la configuración

```
\Configure{newtheorem}{%
  \IgnorePar\NoFonts\HCode{<div id="teorema"><strong>}%
  \EndNoFonts%
}{%
  \NoFonts\HCode{</strong><em>}%
}{%
  \IgnorePar\HCode{</em></div>}\EndNoFonts%
}
```

(obsérvese el par de `\Ignorepar` que se han utilizado para evitar párrafos innecesarios) aplicada a un teorema definido de la forma

```
\newtheorem{thm}{Teoremilla}
```

obtenemos un resultado similar al siguiente:

```
<div id="teorema">
  <strong><a id="x1-24001r1"></a>teoremilla 1. </strong>
  <em>Cuerpo del Teorema...</em>
</div>
```

3.11 Entornos matemáticos

Configuraciones

Las siguientes configuraciones se utilizan para personalizar los entornos matemáticos delimitados por los correspondientes signos.

- `\Configure{[]}{<hook#1>\$\$<hook#2>}{<hook#3>\$\$<hook#4>}`: configuración para el entorno matemático en modo “display” `\[...]`, donde `<hook#1>` es introducido antes del entorno, `<hook#2>` es introducido al comienzo del mismo (justo después de abrirlo), `<hook#3>` justo antes de finalizar el entorno, y `<hook#4>` después de cerrado. Por defecto siempre utilizará imágenes para representar las fórmulas matemáticas.
- `\Configure{()}{<hook#1>\$\$<hook#2>}{<hook#3>\$\$<hook#4>}`: análogo a `\Configure{[]}` pero para el entorno matemático “inline” `\(...\)`. Por defecto siempre utilizará imágenes para representar las fórmulas matemáticas.
- `\Configure{$$$}{<hook#1>}{<hook#2>}{<hook#3>}`: configuración para el entorno matemático en modo presentación, donde `<hook#1>` es introducido antes del entorno, `<hook#2>` es introducido después,

y `<hook#3>` es introducido al comienzo. Por defecto siempre utilizará imágenes para representar las fórmulas matemáticas.

Configurar este entorno es algo delicado, puesto que afecta a otros entornos matemáticos (como `equation`), los que hay que tener en también cuenta, para evitar desbaratarlos y que acaben funcionando incorrectamente. Veamos un ejemplo de lo sensible que puede ser esta configuración con una situación particular:

- `\Configure{<hook#1>}{<hook#2>}{<hook#3>}`: Análogo a `$$` pero para fórmulas “en línea” delimitadas por `$`. Del mismo modo, es algo delicado, puesto que por defecto decide automáticamente si utiliza símbolos o imágenes para representar los distintos elementos de la fórmula “inline”.
- `\Configure{equation}{<hook#1>}{<hook#2>}{<hook#3>}`: Configuración para el entorno matemático `equation`; el primer hook es encajado al comienzo de la ecuación, el segundo es intercalado entre la ecuación y la numeración, y `<hook#3>` es insertado en el momento de finalizar la ecuación.
- `\Configure{PicDisplay}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}`: Configura las inserciones HTML que se realizarán en el entorno `\PicDisplay ... \EndPicDisplay` (ver el siguiente apartado: comandos y entornos). `<hook#1>` y `<hook#2>` son respectivamente las inclusiones para antes y después de la imagen que genera `PicDisplay`. `<hook#3>` se utiliza para insertar el valor de atributo `alt` de la imagen. `<hook#4>` se utiliza para insertar cualquier otro atributo para la etiqueta `img` de la imagen creada.

Ejemplo: la siguiente configuración

```
\Configure{PicDisplay}{%
  \HCode{<div class="image">}%
}{%
  \HCode{</div>}%
}{%
  \PictureFile%
}{%
  class="displaymath"%
}
```

generará una salida HTML para una fórmula matemática similar a la siguiente:

```
<div class="image">
  
</div>
```

- `\Configure{PicMath}{<hook#1>}{<hook#2>}{<hook#3>}{<hook#4>}`: Análogo a `PicDisplay` pero para fórmulas matemáticas entre el texto.

Comandos y entornos

- `\PicMath ... \EndPicMath`: solicita que el contenido matemático inline contenido en dicho entorno sea convertido a imagen.

Ejemplo:

```
\Configure{()}{\PicMath}{\EndPicMath}{}
```

- `\PicDisplay... \PicDisplay`: Análogo a `PicMath` pero para contenido matemático en modo presentación.

4 Referencias

- Documentación oficial de TeX4ht para las configuraciones:
 - <https://www.tug.org/applications/tex4ht/mn11.html#QQ1-11-51>
- Introducción a TeX4ht en la página github de Mika:
 - <https://github.com/michal-h21/helpers4ht/wiki/tex4ht-tutorial>
- Artículos sobre TeX4ht del blog **Blue Danube** de Mr. CVR (cuando está operativa y se puede visitar):
 - TeX4ht-Overview of the process:
 - <http://cvr.cc/?p=297>
 - TeX4ht-Configure Part 1:
 - <http://cvr.cc/?p=323>
 - TeX4ht-Configure Part 2:
 - <http://cvr.cc/?p=362>
 - TeX4ht-Low-level commands:
 - <http://cvr.cc/?p=482>
- Artículo de **Eitan Gurari** (el creador de TeX4ht, fallecido en el 2009):
 - <https://www.tug.org/TUGboat/tb25-1/gurari.pdf>
- Artículo de Maarten Wisse *Hacking TeX4ht for XML output*:
 - <https://www.ntg.nl/maps/28/09.pdf>
- Web de TeX4ht-info con las configuraciones por defecto de TeX4ht:
 - TeX4ht configuration files:
 - <https://www.kodymirus.cz/src4ht/tex4ht-info.html>
 - tex4ht:
 - <https://www.kodymirus.cz/src4ht/tex4ht-infose1.html>
 - latex:
 - <https://www.kodymirus.cz/src4ht/tex4ht-infose2.html>
 - article:
 - <https://www.kodymirus.cz/src4ht/tex4ht-infose5.html>
- Sección del blog de Nasser M. Abbasi para TeX4ht:
 - http://12000.org/my_notes/faq/LATEX/htch3.htm